Vectorization, communication aggregation, and reuse in stochastic and temporal dimensions

Jed Brown *

April 2013

Transformative computing [14, 15] in science and engineering involves problems posed in more than just the spatial domain. Design optimization of unsteady PDE-based models with quantified uncertainty is a prototypical example, requiring iterative methods in spatial, temporal, and stochastic dimensions. Current methods for solving such problems are predominantly based on the concept that the fundamental building block is the solution of a deterministic PDE model, or perhaps one time step of a transient model. This is practical: it permits comfortable partitioning of mathematical analysis and relatively unintrusive software interfaces. This approach encourages algorithms that single-mindedly pursue independence in the stochastic dimension, which is naturally suited to "horizontal" parallel distributions. Unfortunately, this leaves developers of the PDE models banging their heads against the familiar challenges [1, 12] of efficiently utilizing increasingly precious memory bandwidth, hiding and reducing synchronization costs, and obtaining vectorization. Although some of today's methods have been successful at tackling these challenges in the spatial domain, exemplified by the success of spectral element methods, it has proven difficult to balance the competing demands of vectorization, locality, and adaptive resolution for less smooth problems and those with intricate geometry. Meanwhile, the **stochastic and temporal dimensions provide structure that is ideally suited to extreme-scale architectures, if only they could be promoted to first-class citizens**, alongside the spatial dimensions, in algorithmic analysis and in software. Exploiting this structure in "full-space" methods will require crosscutting development [3]: improved convergence theory, efficient hardware-adapted algorithms, high-quality software libraries, and programming tools and run-time systems [6] to facilitate the development of libraries and applications.

Stochastic and temporal dimensions are fundamentally more regular than spatial dimensions: they do not have geometry, have limited forms of boundary conditions, and tend to be smoother. The stochastic and temporal dimensions offer two stages of algorithmic transformation: **aggregation** and **synergy**. The first stage changes the parallel distribution to vectorize over ensembles and to reuse any constant data that may be shared between the ensemble members. It can often be applied without new mathematical developments, though it benefits from new methods to expose parallelism, such as parallel-in-time integrators [8]. We believe the main reason for aggregation in the stochastic and temporal dimensions not being more widespread is lack of programming tools that support maintainable vectorization "across concerns" that are traditionally the managed by different people or different software components. For encapsulation and maintainability, it is important that forward models be developed and debugged without the complexity introduced by these additional dimensions. Some amount of aggregation is necessary for the second stage, for which improved programming tools are even more important. But the true value of synergy across stochastic and temporal dimensions lies in the immense scope for mathematical and algorithmic advances.

*Argonne National Laboratory, `jedbrown@mcs.anl.gov`

Examples

To demonstrate this scope, we review examples of today's successful time-integration and stochastic algorithms that exploit structure in more sequential forms, and highlight more integrative extensions to accelerate convergence, improve vectorization, and reduce communication demands.

Time Integration. The standard solution method [10] for fully implicit Runge-Kutta factors the Jacobian for the first stage, then reuses it (with a spectral decomposition of the method's matrix of coefficients) in a modified Newton method for all stages in the step. Time integration packages using implicit methods such as BDFs either lag the preconditioner in Newton-Krylov methods [11] or lag the Jacobian in modified Newton methods [5]. Rosenbrock-W [10] methods bring this "lagging" concept inside the time discretization, with tolerance for inexactness. In case of a parabolic problem with smooth source term, the "frozen τ" [2] method can be used to obtain the effect of solving all stages without even visiting a fine grid on each stage, but local error estimators still need to compare τ corrections from nearby stages. If we combine this lagging/reuse concept with a time-parallel formulation as permitted by Parareal [9] or spectral deferred correction [7], the solver sees a space-time problem with structured reuse in the time dimension. Reuse of a preconditioner (or matrix) in this context amortizes setup costs such as coarse grid construction in algebraic multigrid and factorization for direct solvers, amortizes memory bandwidth by correcting multiple vectors at once, and amortizes communication latency by solving several vectors at once. The principles behind block and recycling Krylov methods [13] can likely be combined to support the interaction of right-hand sides caused by the interaction of the stage equations. This reuse concept can also be extended to nonlinear solvers [4] and can be incorporated into multilevel solution algorithms. For example, an FAS multigrid method applied to an ensemble of nearby systems might reuse the cell-linearization in a nonlinear smoother, thus vectorizing residual evaluation and correction over all stages. Selective use of multiplicative-in-time smoothers (waveform relaxation) smoothers is also readily available to speed convergence [16] especially in cases where multigrid interpolation in the time direction commits phase errors.

Continuation and Ensembles. In global parameter continuation, parameter estimation, and design optimization, it is common for the difference between nearby problem instances to be representable in a coarse grid or locally-valid reduced-order model, implying that significant fine-grid computation and setup can be reused. Any computation that is not shared across the ensemble (e.g., coarse grids and error estimators) can be easily vectorized, leading to better FPU use, more regular memory access, and fewer messages. The techniques highlighted in the last section can also be used in this context, typically with weaker coupling between ensemble members.

Ramifications

Instead of focusing on parallel methods that emphasize strict independence, we advocate methods that exploit structure to aggregate communication, vectorize, and reuse. By promoting stochastic and temporal dimensions to be on equal footing with spatial dimensions, we expose a myriad of opportunities to relieve the performance and scalability challenges inherent to solving problems in the spatial dimension only. Research priorities to effectively utilize this structure include:

- extending analysis results to "full-space" methods, incorporating available structure in light of plausible performance models for vectorization and communication,
- adaptive recognition of reusability/synergistic structure, and associated load balancing,
- evolution of software interfaces to preserve more nuanced problem structure,
- development of programming tools that allow manipulation of logical vector length and related data structure transformations without significantly complicating the independent code, and while allowing occasional cross-lane operations.

References

[1] J. Ang, K. Evans, A. Geist, M. Heroux, P. Hovland, O. Marques, L. McInnes, E. Ng, and S. Wild, *Report on the workshop on extreme-scale solvers: Transitions to future architectures.* Office of Advanced Scientific Computing Research, U.S. Department of Energy, 2012. Washington, DC, March 8-9, 2012.

[2] A. Brandt and J. Greenwald, *Parabolic multigrid revisited*, Multigrid methods, III (Bonn, 1990), Internat. Ser. Numer. Math, (1991), pp. 143–154.

[3] D. Brown and P. Messina (Chairs), *Scientific Grand Challenges: Crosscutting Technologies for Computing at the Exascale*, 2010.

[4] J. Brown and P. Brune, *Low-rank quasi-Newton updates for robust Jacobian lagging in Newton-type methods*, 2013, *accepted*.

[5] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, *VODE: A variable-coefficient ODE solver*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 1038–1051.

[6] J. Dongarra, P. Beckman, and et al., *The International Exascale Software Project Roadmap*, Int. J. High Perf. Comput. Applics., 25 (2011), pp. 3–60.

[7] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT Numerical Mathematics, 40 (2000), pp. 241–266.

[8] M. Emmett and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Comm. App. Math. and Comp. Sci, 7 (2012), pp. 105–132.

[9] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29 (2007), pp. 556–578.

[10] E. Hairer and G. Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, vol. 14, Springer Verlag, 2010.

[11] A. Hindmarsh, P. Brown, K. Grant, S. Lee, R. Serban, D. Shumaker, and C. Woodward, *SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 363–396.

[12] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelly, S. Scott, A. Snavely, T. Sterling, R. S. Williams, and K. Yelick, *Exascale computing study: Technology challenges in achieving exascale systems.* DARPA, 2008.

[13] M. Parks, E. de Sturler, G. Mackey, D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1651–1674.

[14] R. Rosner (Chair), *The Opportunities and Challenges of Exascale Computing.* Office of Science, U.S. Department of Energy, 2010.

[15] H. Simon, T. Zacharia, and R. Stevens, *Modeling and Simulation at the Exascale for Energy and the Environment*, 2007. See `http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf`.

[16] S. Vandewalle and G. Horton, *Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods*, Computing, 54 (1995), pp. 317–330.